NUMBER 42 DECEMBER 1995

MUJ MACROMEDIA USER JOURNAL

http://www.hyperstand.com

Shock Me

How to Shock Your Web Site from Start to Finish

By Eric Elia

Shockwave, Macromedia's technology for playing back Director movies through World Wide Web pages (see *MUJ*, November 1995, and Hyperstand, http://www.hyperstand.com, for an introduction), has sent seismic ripples through the multimedia- and Web-development communities and raised the ante for creating compelling Web sites. This is the real deal—multimedia on the Web with a reasonable file footprint. And we're not talking about playing a streaming *Doom* across the Internet, but a wider new world of Web design has opened up that may truly separate the men and women from the boys and girls.

Being able to match colored text with a nice background is no longer enough to make you a good Web designer, and the best Web sites will be ones that are Shocked intelligently. So let's get down to business and show you how to Shock your site from start to finish. You may want to follow along on Hyperstand to see Shockwave in action, or visit Macromedia at http://www.macromedia.com for other examples of how the technology is being used. Macromedia has created a superb *Shockwave Developer's Guide* that is available online.

Pieces of the Puzzle

Before you begin to author for the Web in Director, you'll have to take bandwidth into consideration (30KB still takes 30 seconds to download at 14.4Kbps, Shocked or not). Depending on your audience and the pipeline they'll be connecting with, you don't want primary movies that are over 100KB to 200KB. You can link secondary movies to be downloaded in the background and played through the same window using new Lingo commands, which we'll discuss later in the article.

Just as important a consideration is where and how to implement dynamic media to your site. Shocked movies must be downloaded fully before they are played, then they are played within the page. Designers have as much (or as little) layout ability for a movie as they do for an image using the traditional `<IMG SRC>` HTML tag, except Shocked movies require the new `<EMBED SRC>` tag and require width and height specifications. Basically, this means a left, centered or right alignment of your movie in relation to text or regular HTML. When planning movies to place on your page, don't scatter gratuitous animation just for the novelty of it, since each movie uses more RAM. Perhaps more importantly, from a design perspective, too many movies may have the same effect as multiple fonts did at the beginning of desktop publishing. Users felt compelled to exploit the new potential, making their finished product into serious eyesores. In short, use the new technology intelligently.

(continued on page 5)

IN THIS ISSUE...

2 **SPOTLIGHT:** Chuck Henderson explores the new Macromedia Open Architecture.

9 **TIPS AND TECHNIQUES:** Tony Bové looks at high-quality IMA audio for CD-ROM.

11 **TIPS AND TECHNIQUES:** Jonathan Adams shows you how to start building your own programming library.

13 **AUTHORWARE:** Tom King examines Rapid ST, a new prototyping and simulation tool.

16 **VERSION WATCH**

Publisher: HyperMedia Communications Inc. Not affiliated with Macromedia Inc. Macromedia is a trademark of Macromedia Inc.

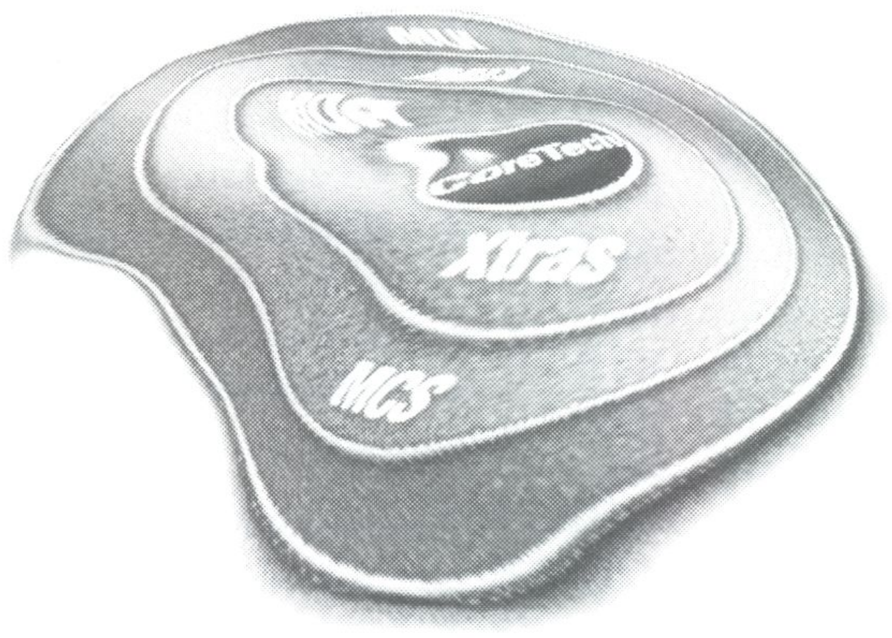
Spotlight

After-Shock

A Closer Look at Macromedia's New Open Architecture

By Chuck Henderson

Perhaps the biggest news coming from the Macromedia User Conference in October was the introduction of Macromedia Studios, a new authoring architecture that integrates all Macromedia applications and, more importantly, opens the door for third-party enhancements of the base functionality of Macromedia applications such as Director. The elements of the Macromedia Studios architecture will start showing up with the next releases of all core products, the most noteworthy of which is Director 5.0 (still scheduled for release in the second quarter of 1996). The highlights of the new Studios concept—the Macromedia Open Architecture (MOA), Macromedia Information Exchange (MIX) and the new Macromedia Common Scripting language (MCS)—are the subjects of this article. (For more on the user conference and Macromedia Studios, see *MUJ*, November 1995, or visit Hyperstand, http://www.hyperstand.com.)



Macromedia's new Open Architecture presents users with a confusing alphabet soup of new acronyms—but they're worth learning.

Gimme MOA

Perhaps the most exciting feature of the Macromedia Studios applications (Director, Authorware, SoundEdit 16, xRes, Extreme 3D, FreeHand, Fontographer and, oh yes, an as-yet-unnamed video editor) is the MOA object. Macromedia Open Architecture is Macromedia's invitation to developers to come build their own little apartment complex, not just next door to but actually adjoining the core structure of Macromedia applications. In fact, the company is providing us with the blueprints (e.g., the application programming interface, or API) to make the docking all the more comfortable.

MOA objects are designed to be dynamically loaded into core applications ("dynamically loaded" means in real time, as requested by the user). They are the fundamental building blocks of Xtras, which are new plug-in extensions that developers will drag into a single Macromedia Xtras folder to augment and enhance and Macromedia applications.

Xtra, Xtra

The future of multimedia authoring is made stronger by the introduction of Xtras. Via MOA, third-party software developers are invited into the Macromedia tent and encouraged to join the feast by creating Xtras, C and C++ routines that, like the familiar XObject, extend the functionality of Director, as well as Authorware, FreeHand, xRes and the other Macromedia Studios applications. Unlike XObjects, though, an Xtra can show up as a menu

MUJ MACROMEDIA USER JOURNAL

http://www.hyperstand.com

EDITORIAL DIRECTOR: Phil Hood
EXECUTIVE EDITORS: Tony Bové and Cheryl Rhodes
MANAGING EDITOR: Eric Elia
CONTRIBUTING EDITORS: Jonathan Adams, David Biedny, Kurt Cagle, Gary Gattis, Chuck Henderson, Tom King, Lynda Weinman
COPY EDITOR: Pat Soberanis
ART DIRECTOR: Nancy Cutler
ASSISTANT ART DIRECTOR: Jake Flaherty
ART INTERN: Agnes Chan
PUBLISHER: Penny Ertelt
ASSOCIATE CIRCULATION MANAGER: Frank Ha

Subscriptions are $175 ($200 outside the U.S.) for 12 issues, $250 ($275 outside the U.S.) for 24 issues. Back issues are available upon request. Single copies are $12, or $6.50 each for 10 copies or more.
Direct all subscriptions, sample copy and back-issue requests to: *Macromedia User Journal*, P.O. Box 469079, Escondido, CA 92046-9779. Voice: (800) 546-6707 (9:00 a.m.–5 p.m. PST) Outside the U.S., call (415) 573-5170, ext. 249. Fax: (619) 743-2590 MUJSub@newmedia.com.
Send article submissions and letters to: *Macromedia User Journal*, 901 Mariner's Island Blvd. #365, San Mateo, CA 94404. MCI Mail and AppleLink: newmediamag; CompuServe: 70640, 2300; America Online: MUJ Editor; Internet: MUJEditor@newmedia.com.
Tony Bové: 30777 S. Hwy, 1/P.O. Box 1289, Gualala, CA 95445. MCI Mail: TBOVE; CompuServe: 70105,722; AppleLink: D0493; America Online: TonyBove; Internet: tony@rockument.com.
Macromedia Director, Macromedia, MediaMaker, Macromedia Three-D, Lingo, Windows Player and XObjects are trademarks of Macromedia Inc.
This publication is not affiliated with Macromedia Inc., nor is Macromedia responsible for its contents.
Macromedia User Journal is published 12 times a year by HyperMedia Communications Inc. Copyright ©1995 by HyperMedia Communications Inc.
(ISSN 1065-3929)

Got a question about your Macromedia User Journal subscription? Call (800) 546-6707 for questions regarding billing and delivery.

selection from within Director (or other application), and can be individualized for Cast, Score, movie or sprite assets. Third parties can now extend the capabilities of Director themselves and build on the core technology, all with Macromedia's blessing.

The most revealing example of a third-party Xtra shown at the user conference was the real-time demonstration of one of the Kai's Power Tools filters. Designed as a Photoshop plug-in, it was invoked as an Xtra from within Director. Within seconds, a filter that looked like a blurred shower door was applied to an ordered group of Cast members from within Director. At play time, the resulting animation showed a clear image emerging from this specialized Xtra blur filter. Once a seemingly closed system, with the possible exception of custom XObjects that worked only at run time, Director is now becoming a dynamic conduit, able to include processes and multimedia objects (now more properly referred to as multimedia assets) from any third party willing to conform to some MOA programming standards.

The development of Xtras is a promising new commercial avenue for advanced Director-based media shops looking for new markets. The new market in this case is made up of the existing and new authoring shops that have already adopted Director and other Macromedia tools. The multimedia market has grown so large in the last couple of years that Macromedia has decided it can accommodate new competition in the form of Xtra developers. This strategy allows Xtra developers to compete with Macromedia in the development of new tools that build upon Macromedia's core technologies. And in so doing, the company's real competition—mFactory's mTropolis, Kaleida/Apple's ScriptX and others—will have a harder time catching up with not just Macromedia, but the extended family of Macromedia Xtra developers.

Into the MIX

Equally provocative was the announcement of the Macromedia Information Exchange (MIX), an MOA-based data integration architecture that will allow users to drag and drop, cut and paste or save and import file types across many Macromedia applications. One application can be launched from another so that, for example, a Director user will be able to double-click on an internal or external sound and immediately launch SoundEdit, manipulate the sound and close SoundEdit, and the changes will be embedded in the Director Cast member.

Similarly, a user will be able to click on a bitmap image in Director, Authorware or FreeHand to launch Macromedia's new image processor, xRes (from the acquisition of Fauve, see *MUJ*, October 1995). And since all Macromedia application interfaces will now look alike (via the Macromedia User Interface, or MUI), the user will feel like the different software components of Macromedia Studios are a single integrated application. Apart from launching one application from within another, users can also embed a data type from another Studios application directly into their run-time productions. For example, Authorware today supports embedded Director movie playback, including full concurrent interactivity. In the future, Director and Authorware will support a FreeHand viewer that allows any FreeHand file to be used as a multimedia asset type, fulfilling the long-vaulted promise of multipurpose media. Multimedia objects will maintain temporal and spatial anchor points when moving from one application to another.

Common Lingo

"Author time," as opposed to "runtime," is a new way to think about the authoring experience. And Macromedia's Common Scripting Language (MCS) will provide a single, cross-product, cross-platform programming solution for the authoring phase as well as for runtime playback. What is typically accomplished using Lingo in Director and user code in Authorware will now be accomplished using MCS, affectionately known as Common Lingo, an enhanced version of Lingo that will provide author-time as well as runtime scripting.

Common Lingo will continue to be used most significantly for developing behaviors for multimedia playback. Because Common Lingo will not be Director specific and instead will be accessible in all Macromedia Studios applications, it can also be extended by defining a new object type or by supporting an existing MOA object type or Xtra. Common Lingo will be able to represent constraints between objects as well, whether they are relationships between real-time 3-D objects in Extreme 3D, or relationships between geometric shapes in FreeHand. Such constraints might include holding a geometric relationship between objects constant or forcing the geometry to conform to a more complex behavior instruction set.

An object is often defined as data wrapped in a procedure. A PICT image file qualifies as an object because it not only contains image data, but is also compressed and includes a header that can convey to an application its "PICT-ness." A more complex object would be a Java "applet" that not only contains data but is wrapped in a full-blown application that can launch itself independently from the calling application and, just as remarkably, destroy itself when the calling application no longer needs it. The distinction between an object and application is further blurred by Common Lingo in the Macromedia Studios environment. Presumably, an Authorware product

could include an object that is a Director movie, and the Director movie could have objects that are FreeHand elements, and all of these could be under the simultaneous control of different Common Lingo scripts.

The macro functionality provided by QuicKeys, a third-party utility that allows complex keystrokes and button events to be scripted into a single keystroke, will now be provided as part of the Common Lingo implementation in all Macromedia Studios applications. It has taken Macromedia too long to realize that script batch control of core applications can be extremely helpful to multimedia developers.

Scripting within Studios applications will also make possible the automatic parsing and formatting of external data as it is imported. For example, a Web site chart maker could import data into FreeHand directly from one Web site and run a script that added appropriate formatting and colors, making an outstanding graphic. The graphic could then be exported into a Director file, "Shocked" with Shockwave and compressed with Afterburner (see Eric Elia's cover article) and uploaded to yet another Web site. Thus, in minutes if not seconds, a text-formatted Dow Jones stock report or other tabular data can be pushed through a carefully scripted pipeline, turned into a compelling graphic and placed back on the Web.

Finally, you may want to extend a Studios application like Director by adding a new menu item or a new modeless panel to control the application itself. Common Lingo will allow such extensions to be accomplished either by calling an MOA Xtra or by writing them directly in Common Lingo. So you could, theoretically, have a Director environment in which pull-down menu items invoke special control windows (like the Cast or Score window) that are really control panels for database Xtras or other features that are completely outside the realm of Macromedia's core technology. In fact, the entire Director environment could become less and less Macromedia and more and more a collaboration of dozens of Xtra applications, chosen and paid for by the user, that populate and comprise the authoring environment.

The Devil in the Details

Because each Macromedia application worries about cross-platform compatibility issues, and because each Xtra sits on top of the Macromedia application (not separate from the application, as is the case with XObjects), developers are able to write Xtras that can be plugged into several Macromedia Studios applications on multiple operating systems. MOA goes beyond the current Microsoft Component Object Model (COM) implementation by providing support for dynamic MOA objects on Windows and Mac systems, as well as support for other operating systems such as OS/2, OS/9, UNIX and others. Afterburner under Netscape is an example of such a cross-platform implementation.

Macromedia will provide MOA interfaces for specific applications in a variety of flavors, the most open ended being the across-all-product interfaces. As in Microsoft's COM, which is the underlying object model of MOA, an interface called IMoaUnknown is the standard interface that queries a MOA object and discovers the interfaces to implement. A set of MOA across-all-product application interfaces will be defined—allocating memory within an application is one example. Developers can write one Xtra that can be invoked by several Macromedia Studios applications by using these interfaces.

Another series of interfaces will be shared across similar products. These include shared pixel data interfaces that are supported by Macromedia Studios applications that handle pixel data (e.g., FreeHand, Director and xRes). Similarly, MOA defines a set of shared multimedia application interfaces that are supported by the multimedia applications in Macromedia Studios. For example, both Director and Authorware support the same shared multimedia interfaces.

Product-specific interfaces define MOA objects that access the internal data structures and major processing loops of each application. These provide public access into the internal hierarchy of objects specific to an application. In Director, a host of MOA objects are defined that give access either to a movie as a whole or to a Cast, an individual Cast member, a Score or a sprite. Again, Xtras can be written to augment the behavior of Director at author time as well as at run time.

An outstanding feature of the MOA interface is the ability to specify well-defined interfaces, such as Photoshop filters, that can abstract and encapsulate application/Xtra functionality. This technique permits the Xtra developer to write "adapter" Xtras for other plug-in technologies. FreeHand 5.5 and Director 5.0 will include an Xtra that accommodates Photoshop-style filter plug-ins. This Xtra scans the disk for Photoshop filters and provides an IMoaPixelFilter "wrapper" for each one found. Any Macromedia Studios application that sees IMoaPixelFilter can use Photoshop filters by dropping in a Photoshop adapter Xtra. With this wrapper technique, using MOA objects is simple compared to supporting Photoshop filters directly.

When an application is launched, any applicable Xtras in the Macromedia Xtras folder are prepared for use. These Xtras are initialized and assigned a reference to the application's call-back class. When the user utilizes an Xtra, the Xtra attempts to invoke an operation on the application by querying the call back to check if the

desired interface is available in the application. If so, the Xtra is free to call whatever method(s) that application interface has defined.

Finally, bear in mind that the same Xtra could be used in Director and Authorware by virtue of polymorphism. That is, when Director launches, it registers any Xtras in the Xtras folder. When the user selects the Xtra, Director selects and initializes it. The same thing happens if the same Xtra is invoked when Authorware is launched. The point is that Director and Authorware will each implement the same Xtra in their own specific fashion, driving the same Xtra without modification. MOA objects by the same name act the same from the Xtra viewpoint; that is, they provide the same socket. But each is implemented differently and independently on the application side.

The times are changing for all of us on the Macromedia bandwagon. The Web, Netscape, Java and Macromedia's Shockwave, MOA, Xtras, MIX and MCS are all becoming invaluable tools for sucking eyeballs away from the TV and hurling them headlong into the multimedia maelstrom. Hold on to your seats and pay attention. Everything will be different in six months.

Chuck Henderson is vice president of research and development for XYZT Corp., makers of application software for the Web. He can be reached at earthmap@mcn.org.

(SHOCK ME: from page 1)

Before Director

First, you'll need to have a few new pieces of software besides Director 4.0.4 and your image editing and audio programs of choice. To convert, or "Shock," a movie to make it Web ready, you'll need Afterburner, Macromedia's cross-platform compression utility. Simply drag and drop a Director (.dir) movie (no Projectors) onto the Afterburner icon, specify a name with the new .dcr suffix and watch it burn. You'll also need the Shockwave plug-in for Netscape 2.0 to view what you've done. These are available online from Netscape and Macromedia.

If you're starting from scratch or modifying an existing movie, you'll need to gather or edit your media. We recommend looking at each Cast member from existing movies to see how you can reduce its file size. Shockwave supports still images and audio files, but not linked media such as QuickTime. Still images should have as small a bit depth as possible without losing too much quality. This can be done in an image editing program or in batches in Equilibrium's DeBabelizer. Macromedia's Afterburner compressor works best with 1-bit (two-color) or 8-bit (256-color) images. If you use 2- or 4-bit images (which you should try to do whenever possible), use Director's Transform Bitmap menu item to convert them to 8-bit. You should not import anti-aliased text or primitive shapes such as squares. These elements can be created in Director and take up little space in an Afterburned movie.

If you feel compelled to include a short QuickTime movie, export the clip as PICTs from an editing program such as Premiere on either platform, or DeBabelizer on the Mac side. DeBabelizer is nice because you can export a batch of images to PICT and rename and scale all of them at the same time. We took a preexisting, flattened QuickTime morph of *NewMedia*'s Awesome Man and Thumps Up icons that was approximately 9 seconds and 1.3MB, exported it to PICT using DeBabelizer and then imported the items into Director. DeBabelizer allowed us to rename the PICTs in sequence. In Director, we sorted the Cast, selected all of the Cast members and used Convert Cast to Time to recreate the animated sequence. The finished, Afterburned movie was 300KB larger than the original QuickTime file, but we were prepared for that outcome, because we had over 100 256-color PICTs. Director wasn't mean to do what QuickTime can, but if you have to convert video, that's one way to tackle the problem.

For audio files, we recommend 8-bit, 11kHz files. Anything lower might be unintelligible, particularly on the Windows side. Anything higher is going to increase your file size significantly. Multiple movies on one Web page

Shockwave: The Real Value

Shockwave is the first real multimedia on the Net. For the next year, at least, it will be the best way to deliver animation and interactivity. In the long run we'll be able to run all kinds of applications directly online, but in the short run Director offers many advantages over other tools. The program is a known commodity—there are 200,000 registered Director users who already know a few things about creating interactive multimedia. That's an advantage it has over Java, JavaScript and other new implementations.

But what will we create with Director that's valuable? The obvious uses are things like games, entertainment or interactive "advertainment." If you want a big payoff for your corporate Web clients, look at ways to save money. Using interactivity on the Web to cut costs can be better than using it to generate revenue, because cost savings go directly to the bottom line. There will be many creative ways to use Shockwave to make your Web site more user friendly, or to support and educate customers. Developers who figure out how to do that will be in high demand.

—Phil Hood

with audio may interfere with each other and slow down other movies, so beware of overdoing it. If you are looping sound to provide an ambient effect, try using lots of little clips and one or two larger files, keeping each element at less than 30KB.

In Director

Shockwave will make Director's Tools palette your best friend. Anything created here—primitive shapes, text or colors—will take up little space in your movie. The entire 448-by-230 movie in Figure 1 was only 13KB when Afterburned. As you can see in the Cast window for this movie in Figure 2, the only graphics were the elements of a bubble animation, head shots of two Hyperstand columnists, a truck that drives across its cell and a rectangle. This empty yellow rectangle (Cast member 1) was created from the Tools palette, resized for each of five rectangular cells and colored red when rolled over. In short, we modified a single Cast member 10 times on the stage. The text was created in Director as well, which allowed us to specify a Mac or Windows system font, color and precise layout—which are difficult in traditional HTML.

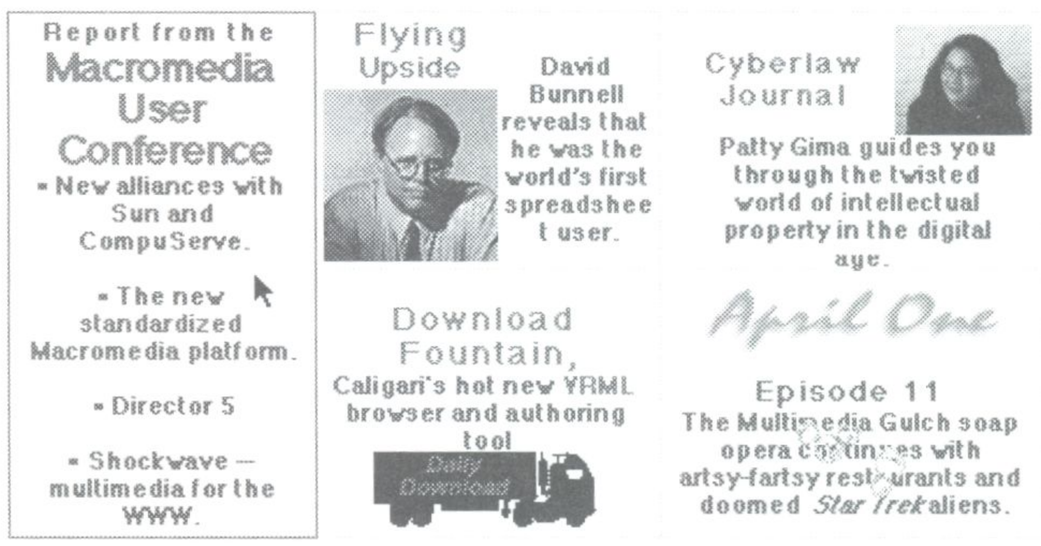

Report from the
Macromedia
User
Conference
• New alliances with Sun and CompuServe.
• The new standardized Macromedia platform.
• Director 5
• Shockwave — multimedia for the WWW.
Flying
Upside
David Bunnell reveals that he was the world's first spreadshee t user.
Download
Fountain,
Caligari's hot new VRML browser and authoring tool
Cyberlaw
Journal
Patty Gima guides you through the twisted world of intellectual property in the digital age.
April One
Episode 11
The Multimedia Gulch soap opera continues with artsy-fartsy restaurants and doomed Star Trek aliens.

Figure 1: The Shocked Hyperstand billboard has simple animation and cell borders that change color when rolled over, and only adds up to 13KB—less than the original GIF files without Afterburner.

Aside from the Tool palette, there are other built-in capabilities you can take advantage of without importing more Cast members. Use the Score's ink effects to change an image's colors without adding to your file size. This is particularly useful for creating visual feedback for rollover buttons and the like. For the time being, one of Director's built-in features to avoid are the transitions. They will interrupt other movies on a single page when they take place. If you have to use a transition, try to use it on a page with only one movie, or keep it brief so it will affect your other movies as little as possible.

As for Lingo, Macromedia introduced a host of new Lingo network extensions. Some Lingo, XObjects, XCMDs

Shockwave Lingo

From Macromedia's *Shockwave Developer's Guide*

New Lingo Network Extensions

Several new Lingo commands provide a Shockwave movie with access to the network. The network is essentially an asynchronous place; that is, it takes time to get things from it. During downloading, a user can continu

Table 1: Features and Commands Disabled in Shockwave

Feature/Command	Description
XObjects	
FileIO XObject	Reads and writes text files that contain information such as user responses, names and addresses, or interactive game information
SerialPort XObject	Makes multiple instances of an object in memory, such as controlling Director movies on two computers from a Director movie on a central computer
OrthoPlay XObject	Controls devices (such as videodisc, videotape and audio CD players) that play video and audio source material
Director-Related Commands	
Resource and XLib commands OpenXLib, OpenResFile, CloseXLib and CloseResFile	Controls resource files located outside a Director movie
Open window and Close window commands	Respectively open and close a window that can play a Director movie
importFileInto command	Replaces a file member with a specified external file
saveMovie command	Saves a movie to a file
printFrom command	Prints whatever is displayed on the stage for a frame or range of frames
System-Related Commands	
Open	Launches and closes specified applications
Quit, Restart and Shutdown commands	Exit Director, restart Director and shut down the computer, respectively
File-name and path properties and functions fileName of cast, fileName of window, getNthFileNameInFolder, moviePath, pathName, searchCurrentFolder and searchPaths	Link to external files, windows, paths and so on
mci command	Passes specified strings to the Windows media control

to interact with a Shockwave movie. Most of the network commands involve starting an operation, then checking to see if it has been completed, and finally getting the results. The Lingo network extensions are different from most Lingo commands, which immediately return the result.

Table 2: Starting Asynchronous Operations

Feature/Command	Description
GetNetText uri	This command starts the retrieval of an HTTP item, to be read by Lingo as text. The uri parameter is a Universal Resource Identifier, which specifies the HTTP item to be retrieved. The HTTP item can be anything, including a Director movie, an HTML page, a graphic and so on. At present, only HTTP URLs are supported as valid uri parameters.
PreloadNetThing uri	This command starts preloading an HTTP item into the local file cache. The uri parameter is a Universal Resource Identifier, which specifies the HTTP item to be referenced. At present, only HTTP URLs are supported as valid uri parameters. In general, an item that has been preloaded can be viewed immediately, since it's taken from the local disk cache rather than from the network. However, it's impossible to determine when an item may be removed from the local disk cache.

NetTextResult, NetMIME and NetLastModDate functions can be called only from the time that NetDone or NetError report that the operation is complete, until the next operation is started. Once the next operation is started, Director discards the results of the previous operation in order to conserve memory.

Each of the functions NetDone, NetError, NetTextResult, NetMIME, NetLastModDate and NetAbort allows an optional parameter specifying the unique identifier of an operation returned by GetLatestNetID.

It's possible to have more than one operation active at a time. When two operations start simultaneously, the Lingo script needs a way to keep them straight during downloading. After one operation starts, and until the next operation begins, the GetLatestNetID() function retrieves a unique identifier for that operation as shown in Table 4.

Table 3: Checking the State of an Asynchronous Operation

Feature/Command	Description
Checking Progress	
NetDone()	This function returns true when the asynchronous network operation is finished. Until that point, it returns false.
NetError()	This function returns an empty string until the asynchronous network operation is finished. Then it returns OK if the operation was completed successfully, or a string describing the error if the operation failed.
Retrieving Results	
NetTextResult()	This function returns the text result of the operation. For a GetNetText operation, this is the text of the HTTP item.
NetMIME()	This function returns the MIME type of the HTTP item.
NetLastModDate()	This function returns the date last modified string from the HTTP header for the item.
Canceling an Operation	
NetAbort	This command cancels a network operation without waiting for a result.

Table 4: Managing Operations

Feature/Command	Description
Checking Progress	
GetLatestNetID()	This function returns a unique identifier for the last asynchronous operation that was started.
Synchronous Commands	
GotoNetMovie uri	This command retrieves and goes to a new Director movie from the network. The new movie occupies the same display area as the calling movie. The uri parameter is a Universal Resource Identifier that specifies the HTTP item containing the movie. At present, only HTTP URLs are supported as valid uri parameters. The URL can specify either a file name or an anchor within a file. For example, both of the following URLs are valid: http://www.yourserver.com/movies/movie1.dcr http://www.yourserver.com/movies/buttons.dc#Contents
GotoNetPage uri	This command opens a URI, whether it's a Director movie or some other MIME type. Using GotoNetPage opens a new page within the net browser.

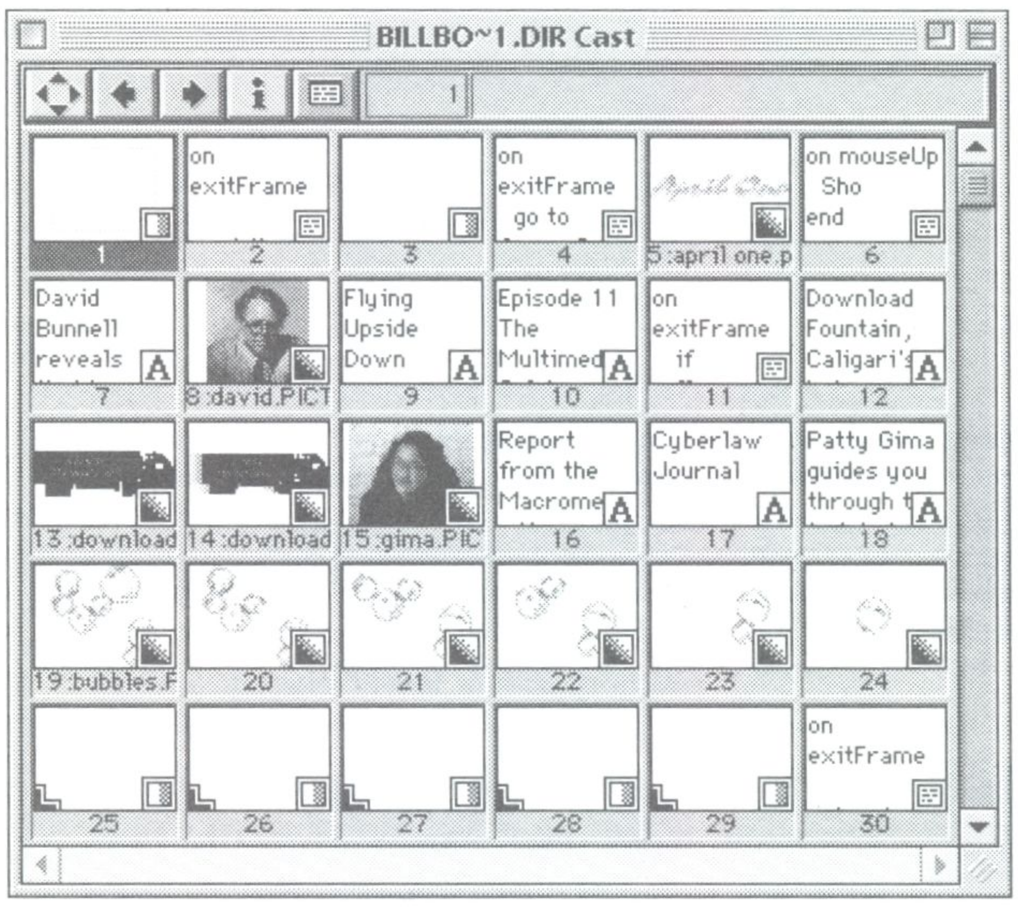


Figure 2: The Cast window for the Hyperstand billboard.

and XFCNs that interact with external software and hardware have been disabled for obvious reasons. Tables 1 to 4, taken from the *Shockwave Developer's Guide,* describe both the new and the forbidden Lingo.

After Director

After your movie is complete, save and compact it in Director to create your .dir file. On the Mac, drag and drop it onto the Afterburner icon; on Windows, open Afterburner, then select the file and save it.

To place your Afterburned movies onto a Web page, you'll need to embed the movies in your HTML document and provide alternate graphics or text for Web users who aren't using the Shockwave plug-in and Netscape 2.0 or another supporting browser (Microsoft, NaviSoft, CompuServe and SGI WebForce are slated to join the party soon).

You'll need the new <EMBED SRC> tag to place the movie in the HTML page. To hide that movie from non-compliant browsers, you'll need a bit of JavaScript, Netscape's new programming language. To provide alternate material, you'll need to use the <NOEMBED> tag, which will be ignored by Netscape 2.0. The following HTML code was used to create the top portion of Hyperstand's home page, as shown in Figure 3.

```
<CENTER>
<SCRIPT LANGUAGE="LiveScript">
<!— Hide this script from non-Navigator 2.0
browsers.
      document.write( '<EMBED WIDTH=480 HEIGHT=95
SRC="http://www.hyperstand.com/shockwave/hyperban-
ner.dcr">' );
<!— Done hiding from non-Navigator 2.0 browsers.
—>
</SCRIPT>
</CENTER>

<noembed>

<CENTER>
<IMG SRC=images/hyperbanner 0923.gif LOW
SRC=images/hyperbanner 0923.jpg ALT=HYPERSTAND>
<BR>
<A HREF=Explore/explore.html><IMG
SRC=images/explorebut.gif BORDER=0></A>
<A HREF=NMToday/today.html><IMG SRC=images/nmt-
but.gif BORDER=0></A>
<A HREF=Awesome/awesome.html><IMG
SRC=images/awebut.gif BORDER=0></A>
<A HREF=Vox/vox.html><IMG SRC=images/voxbut.gif
BORDER=0></A>
<A HREF=Teckno/teckno.html><IMG SRC=images/teck
but.gif BORDER=0></A>
<A HREF=Invision/invis.index.html><IMG
SRC=SITE/images/invisbut.gif BORDER=0></A>
<A HREF=MUJ/muj.html><IMG SRC=images/mujbut.gif
BORDER=0></A>
<A HREF=Download/download.html><IMG
SRCimages/loadbut.gif BORDER=0></A>
</CENTER>

</noembed>
```

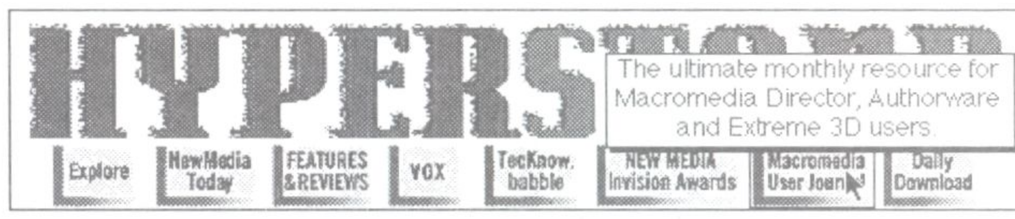


Figure 3: The Shocked Hyperstand banner has animated lightning bolt welcoming audio and pop-up balloon help information for each button.

Aftershock

Keep in mind that the rules to this game change every fe weeks. Visit Macromedia's and Netscape's Web sites often for the latest software, new HTML extensions and more Lingo commands. CompuServe will be supporting Shockwave movies on its online service, and @Home, the new cable modem service for the PC, has announced sup port for the technology as well.

Beyond Director, it is clear that Macromedia will "shock" other important applications such as Authorwar Although nothing has been officially announced yet, Shockwave is really a prototype for a Macromedia Xtra that will work under the new Macromedia Open Architecture (see "After-Shock," page 2). We'll keep you u to date in *MUJ* as developments unfold.

If you aren't planning to move to the Web yet but would like to see your work in cyberspace, *MUJ* welcome any Director movies you'd like us to Shock and post on Hyperstand, mujeditor@newmedia.com

and ounds of almost any length to play with little or no delay on start-up. Interleaving breaks up the sound and video information into small chunks, as shown in Figure 1. In a process called flattening, tools such as MoviePlayer 2.0 interleave a QuickTime movie for playback from CD-ROM.

The non-Red Book formats are easier to work with because you can control the playback of digital sound with standard QuickTime functions. However, the large amount of data that must be processed to play back a stream of CD-quality sound is not compressed, it chokes the throughput of most PCs running Windows, even if the CD-ROM drives are quad-speed.

The basic choice is to sacrifice some of the sound quality for the overall experience (reducing the sampling rate to 22kHz and the bits to 8), or to maintain sound quality by using a compression format for 16-bit, 44.1kHz stereo sound that can also be interleaved. We chose the latter. Of course, our decision, while giving us fantastic sound quality on PowerPCs and properly configured Pentiums, posed all kinds of problems for the remaining PC installed base.

Compress for Play

The compression currently available for CD-quality sound is called IMA 4:1, and it is supported by both QuickTime 2.0 on the Macintosh and QuickTime for Windows 2.01. We supplied both versions of QuickTime on our disc, and integrated QuickTime installation as part of our Setup routine.

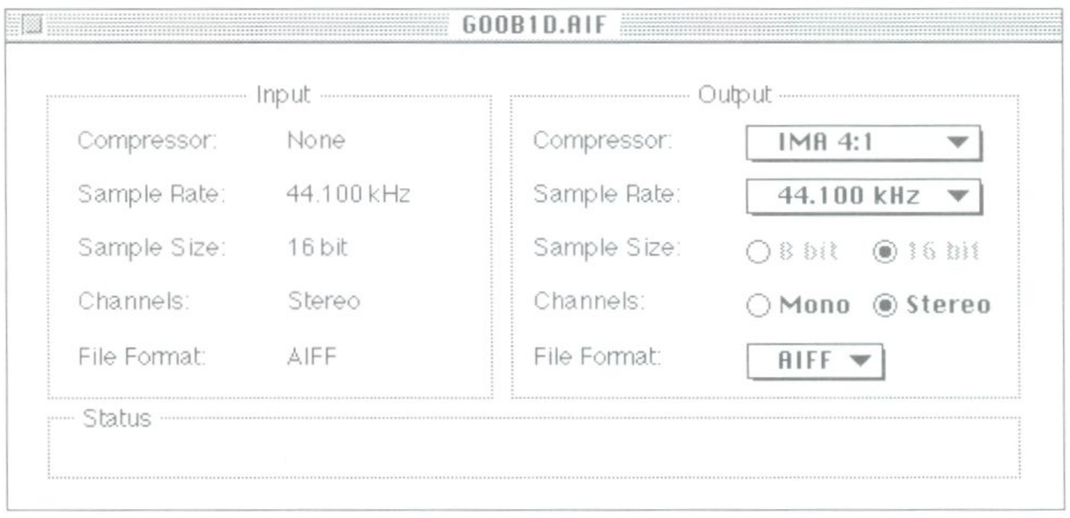


Figure 2: The SoundConverter application on the QuickTime 2.0 Developer Kit CD-ROM lets you save a version of the AIFF sound file in the IMA 4:1 compression format for 16-bit, stereo 44.1kHz sound.

We started with a 16-bit stereo file that was already sampled at 44.1kHz; an audio CD track usually worked nicely. In some cases we transferred material from DAT to this format. This file, stored in the popular AIFF format, can be opened with SoundConverter (supplied on the QuickTime 2.0 Developer Kit CD-ROM). SoundConverter lets you convert an AIFF file to one that is compressed with IMA 4:1.

To compress a high-quality audio stream with SoundConverter (Figure 2), be sure to choose the appropriate sampling rate (44.1kHz in our case), sample size (16 bits), and file format, and the stereo option. Use Save As to save the new file containing the compressed audio.

What can you do with the compressed audio in Director? Practically no other tool in my extensive QuickTime arsenal allows you to edit video and combine video tracks with a track of IMA 4:1 audio. I hope vendors will start to support it. In the meantime, Macromedia Director was the only tool that allowed me to combine video and still images with IMA 4:1 audio.

Using Director to Combine Material

You can do two things in Director with a track of IMA 4:1 audio: You can import it as a single QuickTime movie and use it in a Director presentation, or you can export it with whatever else is on the stage as a QuickTime movie. Both operations are invaluable if you intend to work with this compressed audio format.

For example, we created a full-screen animation sequence that uses a Jefferson Airplane song that is 2 minutes, 30 seconds long. To do this, we imported the compressed audio file into a QuickTime file using MoviePlayer 2.0, saved it, and then, after quitting MoviePlayer, imported the newly created QuickTime file (containing IMA 4:1 audio) into a Director movie and played that.

The result is stunning on PowerPCs and on Windows PCs with 16-bit sound cards: CD-quality sound playing back with animation. And an IMA 4:1 16-bit, stereo, 44.1kHz sound file is one-fourth the size of the uncompressed file—it's approximately the same size as an 8-bit, stereo, 44.1kHz sound file.

Now, what if you want to create a QuickTime movie of images, video clips or both that uses a compressed audio soundtrack? Our method was to export the Director frames to a QuickTime file. In some cases we combined QuickTime video and QuickTime compressed audio files by importing both into an empty Director Movie, arranging them to play on the Stage, and exporting the Stage frames to a single QuickTime movie. Finally, we used MoviePlayer to save a version that is self-contained (flattened) and playable on non-Apple computers, thereby interleaving the file and making it ready for cross-platform use.

Flattening and Data Rates

While all this stuff plays neatly from a hard disk, the challenge of using QuickTime movies in CD-ROM production is to find the appropriate data rate for your installed base of CD-ROM drives.

Many of you know that single-speed CD-ROM drives on Macs can play back QuickTime video streams (with 8-bit audio) if the data rate is around 90KBps or lower.

You may also know that double-speed CD-ROM drives on Macs might be capable (depending on other system characteristics) of playing back video streams smoothly even if they are around 150KBps, or even as high as 180KBps.

You may not know that Windows-based PCs impose even greater restrictions on data rate, largely because video drivers are not standardized and CD-ROM buffers aren't easy to modify. (These problems are alleviated somewhat with Windows 95.) Even if you have a quad-speed CD-ROM drive and a 16-bit sound card, your video card may choke on the data if it comes in at a rate too high to process.

As far as I can tell, with all the testing we've done, Macs and Windows PCs behave differently when confronted with too much data, too soon. Macs will drop frames of the picture to accommodate sound so that the sound plays without interruptions ("hiccups"). Windows will do no such thing—the video is favored, even if it can't keep up (thanks, Microsoft). That means the sound will most likely be interrupted if the data rate is too high.

Now it can also be told that the most popular tools for lowering the data rate (Premiere, MovieShop, etc.) do not work with—you guessed it—IMA 4:1 compressed audio (thanks, Apple).

Thus, you should lower the data rates of your visual QuickTime clips *before* combining them with the IMA 4:1 audio. And in so doing, don't forget to add in the 47.5KBps (or about that) data rate for the compressed audio. We have found that uncompressed 16-bit, 44.1kHz stereo sound needs a data rate of about 174KBps, so 47.5KBps is quite an improvement. You must add in the video data rate to get a total, and that total must be low enough to accommodate your installed base.

You must also flatten the QuickTime movie to interleave the sound and video tracks. We used MoviePlayer 2.0 to do this final step. The exception is when playing an imported QuickTime audio-only movie from within Director or a projector. In that case, you can either load all images into memory before playing the QuickTime movie, or use preLoad commands to control memory usage, simulating your own version of interleaving.

Remember, QuickTime movies mixed with Director movies can push any Windows machine to its limit and force Director (or the projector) to use the swap file, which slows everything to a crawl. That's the last effect you may want for an MTV style production.

Tips and Techniques

Personal Development Library

By Jonathan Adams

One of the ongoing challenges for media developers is to stay on top of trends, maintaining a repertoire of compelling strategies for presenting content. One way to do this is to check out as many interactive media titles and marketing demos as possible while keeping an idea file of presentation and instructional strategies. As you work on various projects, your idea file becomes a source of inspiration as well as a resource for synthesizing and generating ideas.

Although interactive marketing and educational titles tend to be very different in scope and depth, they are similar in their treatments of information presentation, media structures and interface design. Marketing demos can be a wonderful vehicle for forming a sense of your design tastes and sorting out the kind of look and feel you want to present as your personal trademark. In the same sense, education and training titles will help you see how large amounts of information can be effectively segmented and structured.

So, if you want to make your life easier as a programmer, it pays to have a library of examples. The best way to start your library is with simple button handlers, video commands, fileIO or animation effects. As with all libraries, a cataloging system will make items easier to find as your collection grows. Global cataloging schemes for computer-based delivery systems are linear presentations, demonstrations and practice sessions, tutorials, games, simulations and discovery/problem solving. You might further subdivide your examples by function, such as navigation, record keeping, video controllers, initialization scripts and so on.

A Record-Keeping Exercise

This example of an interactive message pad is a strategy for record keeping; it is most appropriate for the discovery/problem-solving category. The goal is to build a message pad that leaves you free to drag your pad to different locations on the screen to make notes or record ideas.

Jonathan Adams (PilotMedia@aol.com) is the principal consultant with Pilot Media Inc. He teaches Instructional Design for Multimedia, Scripting with Lingo, and Director in the University of Massachusetts multimedia certificate program.

Given this scenario, a text field and a button hot spot for automating the message pad will be required. Both the text field and the button will have to behave in harmony with message pad, in terms of their relative position with the message pad when it is dragged.

The difficulty in developing a message pad for your library is to develop the key scripts associated with making the message pad functional, leaving aside situation-specific elements such as saving to disk (see *MUJ*, September 1995, for examples of fileIO), page turning and printing. By reducing the complexity of the library piece, it will be easier to re-adapt in conjunction with other demo files you have built.

With this in mind, it is possible to boil the essential functions of the pad down to the following:

- A single movable sprite created without using "movableSprite of sprite"
- A text field and hot-spot move with the base sprite, maintaining relative positions
- Coding to take into account the annoying cursor errors associated with repositioning text fields
- A means for tracking the position of the message pad
- Assigning an inactive and active location for the message pad
- A means for clearing the text field (see *MUJ*, October 1995, issue for more examples of clearing text fields)

This may sound like a lot to ask for a simple example to be stored in a library, but having this on hand now will save hours of painful development later. Let's take a look at the global variable first.

InitPadLoc. Global variable initialized with on StartMovie; used to store the horizontal and vertical coordinates of the message pad in the inactive position.

The handlers are straightforward; two manage the drag function of the message pad, and one clears the text field.

DragMe. Score script handler called with a mouseDown event; placed on the message-pad sprite.

CheckLoc. Score script handler called with a mouseUp event; placed on an invisible rectangle (hot-spot) sprite.

ClearText. Movie script handler called from `on startMovie` that empties the text fields.

```
on startMovie
   Global InitPadLoc

  —The start movie handler initialize the global
variable with the inactive position of the pad,
and sets the puppet of the message pad channels to
true.
     set initPadLoc = point(301, 364)
     repeat with i = 48 down to 46
        puppetSprite i, TRUE
     end repeat
     ClearText
end StartMovie
```

```
On DragMe
  — The drag manager first determines the dif-
ference between the location of the mouse click
and the location of the registration point of
the message pad and placing this value to a
local variable. By setting the position of the
message pad to the difference between these two
points, the message pad will not "jump" to match
the registration points.
     put the locH of sprite 46 - the mouseH into
Hoffset
     put the locV of sprite 46 - the mouseV into
Voffset
   — By toggling the visible of the textfield
sprite the cursor will update itself in the new
location of the message pad sprite. The other
option is to set the editableText of sprite to
false when the pad starts to move and then to set
the editableText of sprite to true on mouseUp. If
this action is not taken, the cursor will not
travel with the text field.
     set the visible of sprite 47 = FALSE
     set the visible of sprite 47 = TRUE
     updateStage
  —By setting the constraint, the user will not be
able to "toss" the message pad off the stage.
     set the constraint of sprite 46 = 45
  — The stillDown function is used to make the
message pad drag with the movement of the mouse.
The updateStage command redraws the stage as the
message pad sprite moves.
     Repeat while the stillDown
        set the regPoint of cast "textBox" = the
regpoint of cast "messagePad"
        set the locH of sprite 46 = the mouseH +
Hoffset
        set the locV of sprite 46 = the mouseV +
Voffset

        set the locH of sprite 48 = the locH of
sprite 46 + 145
        set the locV of sprite 48 = the locV of
sprite 46 - 15
        set the locH of sprite 47 = the locH of
sprite 46
        set the locV of sprite 47 = the locV of
sprite 46
        updateStage
     end repeat

  End DragMe
```

```
on CheckLoc
   Global InitPadLoc
  —The checkLoc handler is called from the hotspot
which travels with the message pad sprite. The
first order of business here is to get the current
```

```
location of the message pad sprite and store that
information in a local variable.
    put the LocH of sprite 46 into CurrentLocH
    put the LocV of sprite 46 into CurrentLocV

    set PadLoc = point(CurrentLocH, CurrentLocV)
    -- This section makes the comparison between the
starting location and the current location of the
message pad sprite. When the message pad is in a
location other than the start up position, a click
on the hotspot returns it to the starting point.
If in the starting position, a click on the
hotspot brings the message pad into a predeter-
mined location on the stage.
     if PadLoc <> InitPadLoc then
        set the visible of sprite 47 = FALSE
        set the visible of sprite 47 = TRUE

        set the loc of sprite 46 = InitPadLoc
        set the loc of sprite 47 = the loc of
sprite 46
        set the locH of sprite 48 = the locH of
sprite 46 + 145
        set the locV of sprite 48 = the locV of
sprite 46 - 15
        updateStage

     else
        if PadLoc = initPadLoc then
           set the visible of sprite 47 = FALSE
           set the visible of sprite 47 = TRUE

           set the loc of sprite 46 = point(300,
75)
           set the loc of sprite 47 = the loc of
sprite 46
           set the locH of sprite 48 = the locH of
sprite 46 + 145
           set the locV of sprite 48 = the locV of
sprite 46 - 15
           updateStage
        end if
     end if
  end CheckLoc
  ---------------

  on ClearText
  -- Checks the cast for textfields in a range of
castmembers and clears the text
     repeat with i = 1 to (the number of
castMembers)
        if the castType of cast i = #text then
           put " " into field i
        end if
     end repeat
  end cleartext
```

In conclusion, observation and adaptation are the keys to increasing your expertise as a developer. What might appear to the user as a totally new and innovative treatment is most likely the result of some long hours of research and observation. Your library will serve not only to save time in development, but will spark new ideas and provide a source of inspiration and accomplishment. In future installments of this column, I will be presenting treatments and instructional strategies that you might incorporate in your library of working demos.

Authorware

Rapid ST: Simulations and Prototypes

By Tom King

What should a poor author do if an industrial designer wants help in prototyping a new device? Not only will you need to generate a user interface, but you will need to mimic the device's real-world rules, because without them it would be difficult, if not impossible, to assess the usability of the new device. And a simulation that doesn't truly simulate (and simulate promptly) may do more harm than good for training.

Reach for Emultek's Rapid ST, a Windows-based object-oriented simulation and prototyping tool. It works very well by itself or within Authorware, so you can still take advantage of Authorware's power and the skills you've developed with it.

While Authorware excels in media control and in presenting logic in a succinct and easily modified form, and its flow-line metaphor allows anyone to create and manage a complex interactive multimedia presentation, the program's scripting language lacks some functions that complex programming demands, such as reusable, author-defined subroutines and functions. In short, Authorware is great when content is king, but it can be challenging when complexity reigns.

There's a lot of ground to cover before you can really get into Rapid ST. It is a good component of your tool set, providing objects and development tools for creating and maintaining complex simulations of real-world devices and systems.

Rapid was designed for creating simulations from the ground up, and it has built-in support for communication with a variety of authoring systems, especially Authorware. It includes libraries of simulation items such as buttons, switches, displays, dials, indicator lights, storage tanks and other widgets. As a developer, you literally drag and drop these items into place to create your simulation. Just as Authorware has both a Presentation window and a Design window (flow line), Rapid has tools for creating the look of your simulation and tools for con-

trolling its behavior.

Rapid was born out of necessity. Emultek's founders are engineers who had worked on the Israeli Lavi fighter jet in the late 1980s, assisting with interface design for the cockpit. They used a general-purpose programming language to create prototypes that were subject to misinterpretation of both the layout and the logic. Rapid was developed because the designers needed tools to help automate and streamline the process.

The State of State Machines

The engineers began viewing the system as a "state machine" to clarify communication and to facilitate development. This model represents the controls and indicators as combinations of "states." Each state is unique, but it may duplicate elements of other states. Even a simple table lamp and wall switch, as shown in Figure 1, can be represented as a state machine.

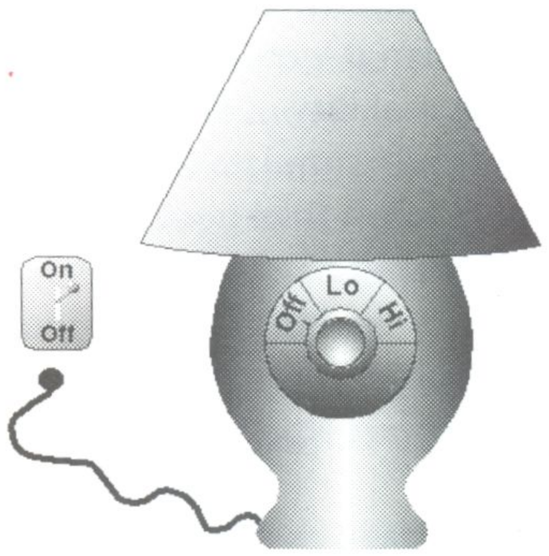


Figure 1: Table lamp at a Holiday Inn.

Moving the wall switch, for example, would trigger a transition from one state (electricity off) to another (electricity on). The terms "trigger" and "transition" are important: Triggers are events or conditions that make the machine move from one state to another. Transitions are the links between states. These transition paths aren't necessarily bidirectional or symmetrical. Some states might not be directly connected. For instance, there are transitions from On to Off and Off to On for the wall switch. However, the lamp switch has transitions from Off to Low, and Low to High, but not Off to High.

The table summarizes all the possible states for a floor lamp with High and Low settings that is plugged into a wall switch. Think of the times you've flicked a light switch and nothing happens. You flick it again, walk over to the lamp, flick the lamp's switch once or twice and then maybe go back to the wall switch again. Imagine writing a flow chart that shows all of the possible combinations to turn the light on or off and adjust the brightness high or low.

Lights Out at Holiday Inn

As the simple hotel-light example implies, representing state logic with procedural flow charts is cumbersome. There are many ways to get to a given state, and all the rules about switch positions and previous positions need to be repeated in the decision flow of each procedural path. State diagrams (or logic charts in Rapid parlance) are more object oriented. State diagrams look a lot like the Venn diagram you may remember from high-school math, except that the Extended State Transition diagram used by engineers not only conveys membership in a "set," but also indicates the transitions and triggers that connect the states when the machine moves from one state to another.

Just as there are many ways to use Authorware's flow line to create a presentation, the same device can be represented by different state-machine models. For example, the hotel room can be modeled from the perspective of lamp or a switch. Modeling it based on the light simplifies the number of modes, as indicated in Figure 2.

Lamp State	Wall Switch	Lamp Switch	Lighting Condition
State 1	On	Off	Out
State 2	On	Low	Dim
State 3	On	High	Bright
State 4	Off	Off	Out
State 5	Off	Low	Out
State 6	Off	High	Out

Figure 2: Rapid logic chart for a state machine based on the lamp states

Figure 2 doesn't really convey much about the physical configuration. Furthermore, the underlying conditions of the transitions are fairly complex. Each transition has to look at both the wall-switch status and the lamp dial. Another way to look at this model would be from the perspective of the switches. With the wall switch as a master power switch, the transition conditions (the triggers) can be simplified to focus on just the wall switch (power

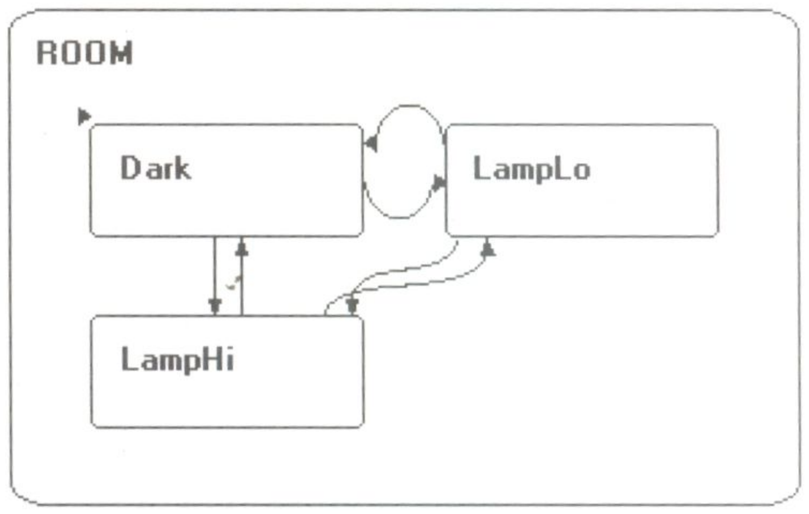


Figure 3: Rapid logic chart for a state machine based on the switch states.

On/Off) or the lamp dial (Off/Low/High). See Figure 3 for a simplified model based on switch status.

When we flip the wall switch, we expect the light to turn on at the High setting. The Rapid model needs to accommodate where the lamp dial was left when the wall switch turned off. The lines from the PowerOn box to the Dial Off to Dial Low states take care of "remembering" where the switch was last. I could have included transitions from one dial state to another, but that would be redundant for two reasons: First, if the power is off, the dial settings are irrelevant. Second, the dial modes are all children of PowerOn. Since they are children of PowerOn, the same triggers are active. Rather than connecting the PowerOn dial states to each other, I let them "inherit" the same transitions and triggers that are activated when transitioning from PowerOff to PowerOn.

A Bird's-Eye View

With all this information under your belt, you will have a better appreciation of Rapid and the different tools or "views" it offers. The first tool you're likely to use is the Layout Editor. Figure 4 shows a sample layout implementing the lamp scenario. Note the palette of objects along the left side. To build this layout I simply dragged a switch, a dial and a lamp from the palette into the work space. Rapid includes all the basic widgets you're likely to need, but you can edit them and make your own as well. These elements are the building blocks of your simulation,

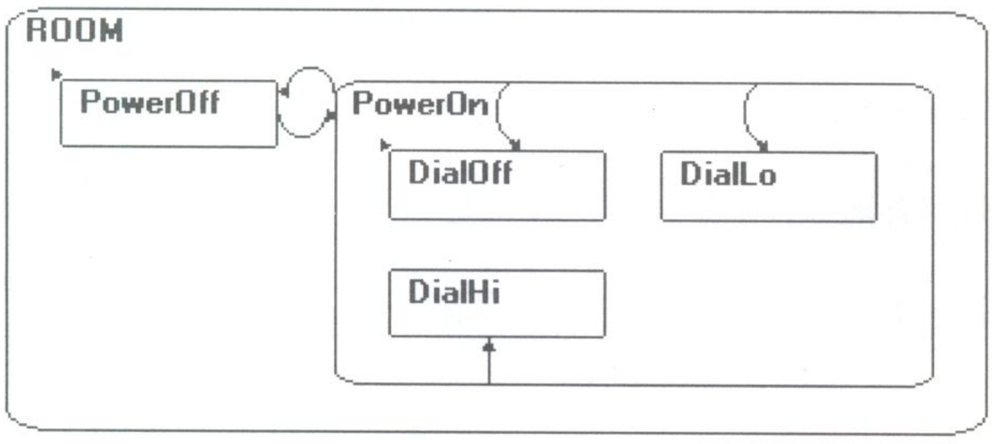


Figure 4: Layout Editor.

Send Us Your Feedback

As part of our effort to improve the *Macromedia User Journal* and give you a resource that suits your needs, we are seeking your input. Please tell us what you like or don't like about the *MUJ*, what you'd like to see more or less of, covered or ignored. Who are the developers you want to learn about? What are the tools you want reviewed? Which techniques do you want demonstrated? And what's the news you can't live without? Feel free to e-mail, write or fax:
MUJEditor@newmedia.com
(415) 573-7446; MUJ Editor, 901 Mariner's Island Blvd., Suite 365, San Mateo, CA 94404.

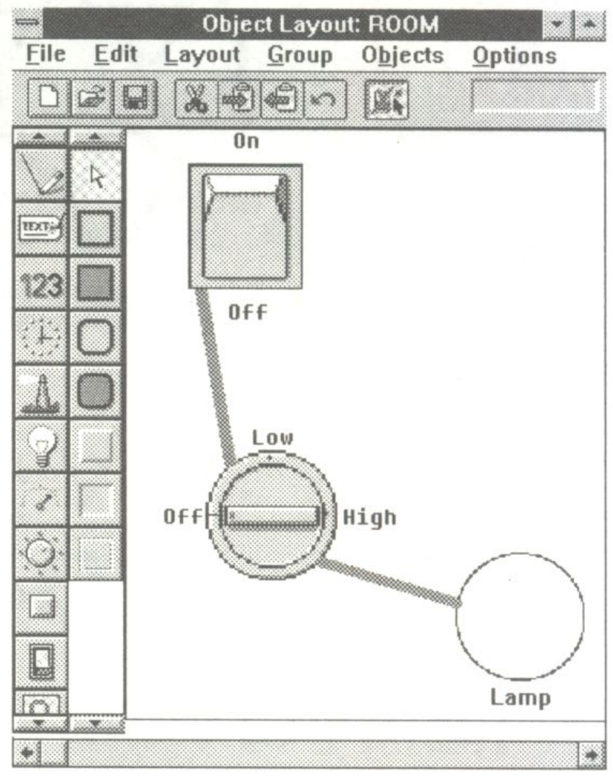


Figure 5: Mode Tree.

and the Layout Editor lets you label them and create the necessary backgrounds.

After you've created the layout for your simulation, you'll need to begin creating modes or states. You do this

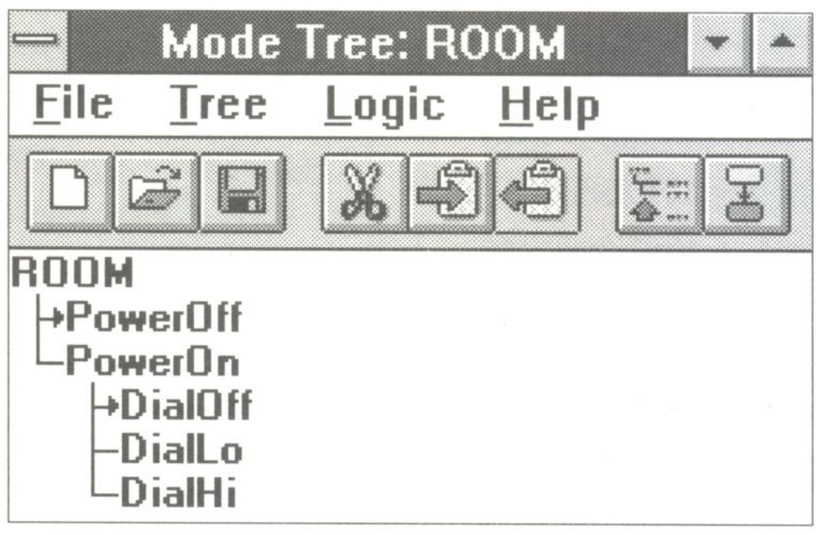


Figure 6: Logic Editor and Mode Browser.

using the Mode Tree, shown in Figure 5. Finally, to create the triggers, transitions and events that bring your Frankenstein to life, you will use the Logic Editor and the Mode Browser, shown in Figure 6.

That's it, almost. It's as easy as 1 (Layout), 2 (Modes), 3 (Logic). Nevertheless, a few more subtleties need to be clarified, and we still need to know how all this can hook into Authorware. Tune in to the next *MUJ* for some sample code and to see how Rapid can:

- Run your simulation in Free play mode all by itself;
- Run concurrently with Authorware and communicate with Authorware;
- Run concurrently with Authorware and be controlled by Authorware.

Tom King works at Solis Design Inc. in San Mateo, California. He can be reached via e-mail at SolisInc@aol.com.

Version Watch

December 1995

Product	Platform	Version	Price
Application Software			
Authorware (Includes Director Studio bundle)	Mac	3	$4,995
Authorware (Includes Director Studio bundle)	Win	3	$4,995
Action!	Mac	1.0.4	$199
Action!	Win	3.0	$199
Action! CD-ROM Edition (Action! SoundEdit Pro, ClipMedia 3)	Mac	—	$299
Action! CD-ROM Edition (Action! Turtle Beach's Wave Tools, ClipMedia 2)	Win	—	$299
Director ($29 upgrade for registered 4.0 Mac users)	Power Mac	4.0.4	$1,195
	Win	4.0.4	$1,195
Fauve xRes	Mac	1.2	$799
Fontographer	Mac	4.1	$495
Fontographer	Win	3.5	$495
FreeHand	Mac	5.5	$595
FreeHand	Win	5.0	$595
MacroModel	Mac	1.5	$895
MacroModel (includes Pixar's MacRenderMan)	Mac	1.5	$995
MacroModel (includes Macromedia Three-D)	Mac	1.5	$995
MacroModel (includes Pixar's Windows RenderMan)	Win	1.5	$995
SoundEdit 16 (upgrades from SoundEdit, SoundEdit Pro and MacRecorder Sound Systems are $119; SoundEdit 16 replaces SoundEdit Pro and other earlier versions of SoundEdit)	Mac	1.0.1	$379
X-Objects Developer's Toolkit (Windows version is included with Director for Windows 4.0)	Mac	N/A	$50

Director Studio bundle, $1,995 for Macintosh or Windows, includes Director, MacroModel, SoundEdit 16 (Mac), SoundForge (Win), and Adobe Premiere.

Graphic Design Studio bundle, $995 for Macintosh or Windows, includes FreeHand, MacroModel, Fontographer, Pixar RenderMan and Fractal Design Painter.

ClipMedia			
ClipMedia 1: Business & Technology	Mac	2.0	$395
ClipMedia 2: Industry at Work	Mac	2.0	$295
ClipMedia 3: Perfect Presentations	Mac	2.0	$195
ClipMedia 1: Business & Technology	Win	2.0	$395
ClipMedia 2: Industry at Work	Win	2.0	$295
ClipMedia 3: Perfect Presentations	Win	2.0	$195

Coming Soon in the MUJ

- Part 2 of Tom King's examination of Rapid ST, and a first look at Designer's Edge, Allen Communication's new pre-authoring tool for Authorware, Quest and IconAuthor users.
- New year, new products. A sneak peek at Macromedia's first rev of xRes and an in-depth look at Extreme 3D.
- Top-secret Authorware news. We'd tell you now, but we'd have to, you know…
- Tony Bové shows you how to master CD-ROM load time and digital video space with Director.
- Kurt Cagle explains how the new Macromedia Open Architecture and Common Scripting will affect the Lingo you know and love.